Spatial iterative learning control for robotic path learning

Article (Accepted Version)

Spatial Iterative Learning Control for Robotic Path Learning

Lin Yang, Yanan Li, *Senior Member, IEEE,* Deqing Huang, *Senior Member, IEEE,* Jingkang Xia, and Xiaodong Zhou

***Abstract*—A spatial iterative learning control (sILC) method is proposed for a robot to learn a desired path in an unknown environment. When interacting with the environment, the robot initially starts with a predefined trajectory so an interaction force is generated. By assuming that the environment is subjected to fixed spatial constraints, a learning law is proposed to update the robot's reference trajectory so that a desired interaction force is achieved. Different from existing ILC methods in the literature, this method does not require repeating the interaction with the environment in time, which relaxes the assumption of the environment and thus addresses the limits of the existing methods. With the rigorous convergence analysis, simulation and experimental results in two applications of surface exploration and teaching by demonstration illustrate the significance and feasibility of the proposed method.**

***Index Terms*—Path learning, sILC, teaching by demonstration, surface exploration, learning law.**

I. Introduction

There is an inevitable trend that robots are moving out from behind cages and coming to interact with surrounding environments, due to extensive applications that require the robots to get in physical contact with passive environments, e.g., grasping. In addition, the need for human-robot interaction, where a human is an active environment from the robot's point of view, is becoming cumulatively urgent with the quick aging of societies over the world. On the one hand, the combination of human intelligence and robot power has been found effective in complicated situations where robots alone cannot fulfil task requirements [1]. On the other hand, robots are anticipated to learn from short-term and long-term interactions with humans [2]. In a typical interaction task such as luggage loading and offloading in airports, a robot can be programmed to follow a predefined task trajectory, and at the same time it should be able to adjust its trajectory by a human worker when necessary, e.g., to avoid an unexpected obstacle. Conventional industrial robotic technologies are far from responding to this need, as preprogrammed control structures to complete a single task do not allow real-time human interventions. New robots should be partners who can integrate into a human's normal work and life, and have certain autonomous behaviors [3]. By collaborating with and learning from human partners, robots are capable of reducing labor costs, providing high productivity and accomplishing tasks in hazardous environments [4].

Such a direction towards robots' learning through robot-environment interaction requires the development of novel robotic learning techniques. In the field of force and impedance control, various learning and adaptive algorithms have been developed, such as [5], [2]. However, these works require time-invariant environmental parameters. Iterative learning control (ILC) has been explored for time-varying environmental parameters [6], [7], but these parameters are assumed to be periodic in time.

In the field of human-robot interaction, a wide variety of tasks can be demonstrated to a robot by manually guiding either a tool or the robot's end-effector through a haptic interface. Many works have used an idea of processing raw data collected in demonstration. In [8], by measuring the forces exerted by a human operator and the positions of a robot's end-effector, data analysis is conducted in order to understand what the operator intends to do. Then, necessary information is obtained to generate the hybrid control program, that tells which control mode, position or force, should be taken in each direction, how much force should be exerted and what trajectory the end-effector should follow. Clearly, the problem with this method is that any minor change in the task at hand, such as a shift in the goal position or the avoidance of an environmental constraint along the path, necessitates further demonstrations, and cannot be adjusted in the previous trajectory [9]. Programming by demonstration (PbD), which is also known as learning from demonstration (LfD) or learning by imitation, has emerged as one of the most promising solutions for effective programming of robotic tasks [10]. It is arguably easier for humans, including non-robotic experts, to input poses by moving a robotic arm manually than through coding in a specific form [11]. Trajectory learning is a fundamental component in robotic learning through PbD, of which the purpose is to transfer complex trajectories to robots from humans [12], [13]. Trajectory learning can be used to assist human actions in various fields, e.g., to help patients with physical rehabilitation training [14], [15] or assist surgeons by performing specific subtasks [16]. In these two examples, based on trajectory learning, the rehabilitation action can be continuously adjusted according to a patient's personal preferences and surgical robots can reduce the duration of a doctor's operations. Other methods of trajectory learning are found in [17], [18], [19], which are feasible for completing obstacle avoidance tasks.

*This work is supported in part by National Natural Science Foundation (NNSF) of China under Grants No. 51805025, No. 61773323, No. 61603316, No. 61433011 and No. 61733015.

L. Yang, D. Huang and J. Xia are with the School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, P. R. China. Email: elehd@swjtu.edu.cn

Y. Li is with the Department of Engineering and Design, University of Sussex, Brighton BN1 9RH, UK.

X. Zhou is with the Beijing Institute of Control Engineering, China Academy of Space Technology, Beijing 100094, P. R. China.

However, for some tasks, such as drilling and carving, a desired contact force between the robot and the environment is required, so aforementioned PbD methods cannot guarantee a good control performance. The focus on learning force-based skills, rather than position-based skills, without requiring detailed geometric information, dramatically increases the functional capabilities of our envisioned intelligent robotic tools and assistants. [20] proposes an approach that has two learning phases, which include mapping of the human posture and learning of the interaction rule by using the observed human gesture based on this mapping result. [21] presents a method to optimize the robot's reproduction of the task, which is validated in an experiment with a humanoid robot learning simple manipulation tasks through a kinesthetic interface. [22] develops a method to transfer assembly skills to robots by observing a human-performed demonstration of the skill. An apprenticeship learning approach is proposed in [16], by recording a set of trajectories using human-guided back-driven motions of the robot. These are then analyzed to extract a smooth reference trajectory, which executes gradually increasing speeds using a variant of ILC.

ILC is a suitable control strategy for trajectory learning, because it can update the robot's trajectory in an iterative manner [23]. However, similarly as in the literature of force and impedance control, most of the existing works study trajectory learning based on a time period. However, in many practical applications, learning period is usually time-varying. Let us consider a typical scenario of human-robot collaborative manipulation, in which the robot and its human partner respectively hold two sides of an object. In the first instance the robot passively follows its human partner who applies a force onto the object and moves it to his/her target position. Then, in an iterative manner, the robot learns to adjust its own reference trajectory to reduce the human partner's control effort, and eventually the robot takes full control of the task and frees the human partner. If the robot's learning is time-based, we need an assumption that the human partner can repeat the motion in different trials, but this assumption is difficult to meet due to humans' varieties, e.g., different velocities in each trial. Nevertheless, as the spatial constraints in this scenario remain the same in each trial, it becomes possible for a robot to learn the human partner's spatial skills. Based on this idea, this paper will introduce a novel approach for trajectory learning based on spatial iterative learning control (sILC) [24]. While sILC has been studied mainly in the field of motion control, the present study introduces it to robotic spatial learning for the first time. It will be shown that this approach is effective as the human partner's speed in each trial can be arbitrary, which addresses potential issues associated with the aforementioned fact that it is difficult for the human partner to maintain the same speed for demonstration each time.

Although existing works have introduced iterative ideas into robot control [25], it differs from this article in that it iteratively adjusts the feed-forward force and reference point. The basic idea of the "Incremental Imitation Learning" or "Incremental Kinesthetic Teaching [26]" method in other literatures is to run multiple trajectories, and optimize the new trajectory by comparing with the previous records. These methods are useful for task optimization, but not for local modification of tasks.

In the rest of the paper, we first formulate the problem under study in Section II, by analyzing the system dynamics and listing related remarks. In Section III, we elaborate the proposed method and rigorously analyze the system performance. In Section IV, we present simulation results of a robotic surface exploration task. In Section V, we present experimental results of a LbD task. At last, we draw conclusions of this paper and discuss possible future works.

II. Problem formulation

In this paper, we consider a system where a rigid robot arm physically interacts with an unknown environment. The interaction force between the environment and the robot arm can be directly measured with a force sensor, or the joint force / torque can be measured with a joint sensor and then converted into the required contact force.

A. Robot and environment

According to [27], the dynamics of a n-degree-of-freedom (n-DOF) robot in the operational space are given by

$$M(q)\,\ddot{X} + C(q,\dot{q})\,\dot{X} + G(q) = u - F(t) \tag{1}$$

where X is the position of the robot and q is the vector of joint angle. u is the $n \times 1$ vector of control inputs and $F(t)$ is the measured $n \times 1$ vector of interaction force. $M(q)$ is the $n \times n$ symmetric positive definite inertia matrix, $C(q,\dot{q})\dot{X}$ is the $n \times 1$ vector of Coriolis and Centrifugal force and $G(q)$ is the $n \times 1$ vector of gravitational force.

As an environment can be characterized (locally) by its viscoelasticity, the following simplified interaction force model is employed:

$$F = K_e(X - X_0) \tag{2}$$

where K_e and X_0 are unknown stiffness matrices and the rest position of the environment, respectively.

Remark 1: In the case of human-robot interaction, K_e is the human arm stiffness and X_0 is the human partner's target position.

These parameters are determined by the environment and they are usually unknown to the robot arm. As we consider the robotic learning by repeating the interaction, we can assume that the environment parameters are periodic with a spatial distance S:

$$K_e(s+S) \equiv K_e(s), \quad X_0(s+S) \equiv X_0(s) \tag{3}$$

where s is the spatial coordinate in the contact-free direction. The periodicity of the environment parameters is a realistic assumption for a repeatable interaction task, e.g., the surface exploration presented in the simulation of [7].

Remark 2: In comparison to [7], where the properties of the environment surface are the same for every session along the time axis, in Eq. (3) the parameters are periodic along the spatial axis. As discussed in the Introduction, spatial periodicity is a more relaxed assumption than temporal periodicity. In the surface exploration task, temporal periodicity indicates that

the robot has to move in a same speed in different iterations which is infeasible. In the human-robot interaction, temporal periodicity indicates that the human partner has to repeat their motion with a same speed, which is unlikely to guarantee due to human varieties. Therefore, the existing methods in the literature that are time-based are not applicable to the environment under study in this paper.

B. Robot controller

From Eq. (2), it is found that $F = 0$ when $X = X_0$. In human-robot interaction, it means that the human's target position X_0 is reached. Therefore, a force controller can be developed for achieving $F = 0$. In a more general case, one may expect the robot to maintain a desired force on the environment, e.g., surface exploration. To describe this behaviour, we consider that the control objective is

$$F = F_d \tag{4}$$

where F_d is a predefined desired contact force. From the force model (2), we assume that there exists a desired trajectory X_d yielding F_d, as below

$$F_d = K_e(X_d - X_0) \tag{5}$$

By deducting (2) by (5), we obtain the contact force error equation:

$$E_f = K_e(X - X_d) \tag{6}$$

where $E_f = F - F_d$ is the force error. Then, the control objective becomes

$$E_f = 0 \tag{7}$$

To achieve this control objective, we design the following controller

$$\begin{aligned} u = {} & M(q)\,\ddot{X}_r + C(q,\dot{q})\,\dot{X}_r + (1-v)C(q,\dot{q})\,\dot{E} + G(q) \\ & -KE + F_d \end{aligned} \tag{8}$$

where $E = X - X_r$ is the position error with X_r being the reference trajectory. K is a symmetric positive-definite matrix having minimal eigenvalue $\lambda_{min}(K) > 0$. v is the speed of the robot in the contact-free direction. By substituting the controller u into Eq.(1), the closed-loop system dynamics become

$$M(q)\ddot{E} + vC(q,\dot{q})\dot{E} + KE = -E_f \tag{9}$$

Eq. (9) has a same form of the impedance model in impedance control [28], so the approach to be introduced in the rest of the paper can be similarly developed based on impedance control, as will be explained later.

Remark 3: In order to avoid the uncertainty of human partners in the iteration of the time period, the scene has the nature of space period. System dynamics will be converted from the time domain to the space domain.Therefore, the speed v in the non-contact direction is introduced. This article only discusses the iteration in a single direction. When the $X-Y-Z$ plane iteration is required, v is the component in that direction.

The algorithm proposed in this paper can be used for any curve in the space coordinate system, by applying force / moment in a single direction to change its trajectory.To facilitate the analysis, we only consider the system dynamics in a single direction. In particular, we use $m, c, k, x, x_d, x_r, f, f_d, e, e_f, k_e$ and x_0 to represent a component of $M, C, K, X, X_d, X_r, F, F_d, E, E_f, K_e$ and X_0, respectively. Therefore, according to Eq. (9), the closed-loop dynamic model of the robot arm in a single direction is given by

$$m\ddot{e} + vc\dot{e} + ke = -e_f \tag{10}$$

where the dependence of variables on their arguments are omitted.

III. Path learning

In this section, the approach of path learning will be developed, where we use "path" to replace "trajectory" to indicate the learning independent of time-periodicity. To utilize the spatial periodicity of the environment, the system dynamics will be first transformed to the space domain from the time domain. Then, a learning law will be proposed to update the robot's path. It will be shown that this approach will lead to successful force tracking after a certain number of trials.

A. Transformation from time domain to space domain

First of all, we discuss the relationship between the spatial and temporal coordinates. As have been defined before, t is the time, s is the displacement of the robot in the contact-free direction and $v = \frac{ds}{dt}$ is the speed of the robot in this direction. Therefore, we can obtain

$$\frac{d}{dt} = \frac{d}{ds}\frac{ds}{dt} = \frac{d}{ds}v \tag{11}$$

The spatial differentiator, or the $\bigtriangledown$ operator [24], is defined below and linked to the temporal differentiator:

$$\bigtriangledown = \frac{d}{ds} \tag{12}$$

In order to facilitate the conversion between t and s, let us analyze the relationship between the temporal coordinate t and spatial coordinate s. From $ds = vdt$ we have $s = \int_0^t v(\tau)d\tau$. When the robot's speed $v > 0$, s is a strictly increasing function of t, hence the relationship between t and s is bijective. The function $s = f(t)$ is analytic and the inverse function $t = f^{-1}(s)$ exists globally. Therefore as a variable, $v(t)$ can also be expressed as a spatial function $v(f^{-1}(s))$. Throughout this paper, we make the following assumption.

Assumption 1: The contact-free movement of the robot is in one direction and $v > 0$.

Then, according to Eq. (10), we obtain the closed-loop dynamics in the space domain

$$\begin{aligned} & mv \bigtriangledown e_v(s) + vce_v(s) + ke(s) = -e_f(s) \\ & e_v(s) = \dot{e}(s) \end{aligned} \tag{13}$$

Please note that in the rest of this article, all variables that have not been marked with an argument will default to variables that are functions of s.

Remark 4: The results of spatial iterative learning show that for robot systems, the running direction and running speed are the most important characteristics. When the rotation speed occasionally crosses zero but most of the time it runs at a non-zero speed, we can simply switch to another controller at a speed close to zero, and return to periodic adaptive control when the speed is close to the operating point . If the rotational speed frequently crosses zero, the angular displacement of the rotating mechanism is unlikely to exhibit any periodic behavior. In this case, periodic adaptive control is not suitable.

The situation of $v < 0$ and $v > 0$ is symmetrical and will not affect the stability of the system. For the case of $v = 0$, there are multiple processing methods, because one position corresponds to multiple moments, Eq. (13) is no longer applicable, we consider a simpler processing method: because the scenario we consider is that the path changes periodically in space, Therefore, v = 0 will not always exist in the s field, so we do not iterate at v = 0, and keep the position the same as the previous moment. When there is always v not equal to 0 at this position, iterate normally.

B. Learning law

According to the previous analysis, the control goal of this paper is to make $e_f = 0$ in Eq. (13), and the reference trajectory x_r needs to be obtained by designing the controller. The design of x_r uses $e'_f(s)$, $e_v(s)$ and $e_f(s)$.

To develop the learning law, we rewrite Eq. (10) as

$$x = x_r - e'_f \tag{14}$$

where the signal e'_f is a filtered signal of e_f, which satisfies

$$m\ddot{e}'_f + vc\dot{e}'_f + ke'_f = e_f \tag{15}$$

Eq. (6) in a single direction can be written as

$$e_f = k_e(x - x_d) \tag{16}$$

By substituting (14) into (16), we obtain

$$\frac{e_f}{k_e} = x - x_d = x_r - e'_f - x_d \tag{17}$$

Then, we design the robot's reference trajectory

$$x_r = e'_f + \hat{x}_d \tag{18}$$

where $\hat{x}_d$ is the estimate of x_d. According to Eqs. (17) and (18), we have

$$\frac{e_f}{k_e} = \tilde{x}_d \tag{19}$$

where $\tilde{x}_d = \hat{x}_d - x_d$. From this equation, we can clearly see that $e_f = 0$ if $\tilde{x}_d = 0$, i.e. , if x_d is known, the control objective can be achieved with $\hat{x}_d = x_d$. However, x_d is unknown due to unknown K_e and X_0 in Eq. (2). Nevertheless, we know that x_d is periodic with S, as k_e and x_0 are periodic with S, i.e.,

$$x_d(s + S) = x_d(s) \tag{20}$$

In the following, we utilize the spatial periodicity of x_d to design $\hat{x}_d$ in Eq. (18) to make $\lim_{s\to\infty} e_f = 0$. In particular, we develop the following learning law:

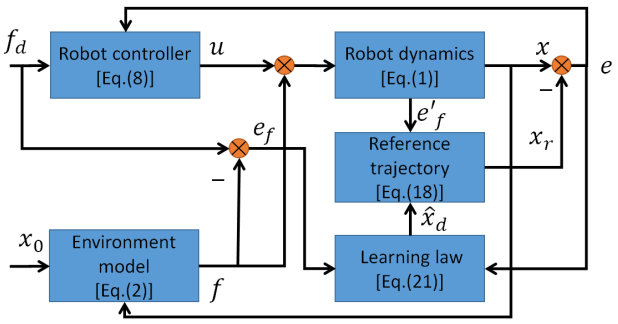

Fig. 1: Block diagram of the proposed control framework

$$\hat{x}_d(s) = \begin{cases} -\lambda e_f(s) + \frac{1}{v}e_v(s), & 0 \le s < S. \\ \hat{x}_d(s - S) - \lambda e_f(s) + \frac{1}{v}e_v(s), & S \le s < \infty. \end{cases} \tag{21}$$

where λ is a positive scalar.

With $\hat{x}_d(s)$ updated in Eq. (21), the reference trajectory x_r is obtained as in Eq. (18). Then, the controller in Eq. (8) can be computed. Note that the trajectory x_r is updated iteratively while the controller u is calculated continuously in time. The structure of the proposed controller is illustrated in Fig. 1.

C. Stability analysis

In this section, we show that the control objective is achieved with the proposed learning algorithm, which is summarized in the following theorem.

Theorem 1: Considering the closed-loop dynamics described by (17), the designed reference trajectory (18) with the updating law (21) guarantees the following results:

(i) the interaction force error asymptotically converges to 0 as $s \to \infty$, i.e. ,$\lim_{s\to\infty} e_f(s) = 0$ and

(ii) all the signals in the closed-loop system are bounded.

Proof 1: The following proof of Theorem 1 is given based on Lyapunov theory. In particular, let us consider a Lyapunov function candidate as below

$$\begin{aligned} V &= V_1 + V_2 \\ V_1(s) &= \begin{cases} \int_0^s k_e[x(\tau) - x_d(\tau)]^2 d\tau, & 0 \le s < S. \\ \frac{1}{2}\int_{s-S}^s k_e[x(\tau) - x_d(\tau)]^2 d\tau, & S \le s < \infty. \end{cases} \\ V_2(s) &= \frac{1}{2}[e_v(s)me_v(s) + k_e e^2(s)] \end{aligned} \tag{22}$$

For $0 \le s < S$, considering Eqs. (16) and (18), the derivative of V_1 with respect to s is

$$\begin{aligned} \bigtriangledown V_1 &= k_e[x(s) - x_d(s)]^2 \\ &= [x(s) - x_d(s)]e_f(s) \\ &= x(s)e_f(s) - x_d(s)e_f(s) \\ &= \hat{x}_d(s)e_f(s) - x_d(s)e_f(s) \\ &= [-\lambda e_f(s) + \frac{1}{v}e_v(s)]e_f(s) - x_d(s)e_f(s) \\ &= -\lambda e_f^2(s) + \frac{1}{v}e_f(s)e_v(s) - x_d(s)e_f(s) \end{aligned} \tag{23}$$

Taking the derivative of V_2 with respect to s, we obtain

$$\bigtriangledown V_2 \quad = e_v(s)m \bigtriangledown e_v(s) + e_v(s)\tfrac{k_e}{v}e(s) \tag{24}$$

According to the closed-loop dynamics (13), the above equation can be written as

$$\begin{aligned} \bigtriangledown V_2 &= e_v(s)(m \bigtriangledown e_v(s) + \frac{k_e}{v}e(s)) \\ &= e_v(s)(-\frac{1}{v}e_f(s) - ce_v(s)) \end{aligned} \tag{25}$$

Therefore ,for $0 \leq s < S$, according to Eqs. (23) and (25), we have

$$\begin{aligned} \bigtriangledown V &= \bigtriangledown V_1 + \bigtriangledown V_2 \\ &= -\lambda e_f^2(s) - x_d(s)e_f(s) - ce_v^2(s) \\ &= -\lambda e_f^2(s) - ce_v^2(s) - x_d(s)e_f(s) \end{aligned} \tag{26}$$

Because $e_v(s)$, $x_d(s)$ and $e_f(s)$ are bounded for $0 \leq s < S$, so $\bigtriangledown V$ is on $0 \leq s < S$ For bounded. Since $\bigtriangledown V$ is bounded for $0 \leq s < S$, V is bounded for $0 \leq s < S$.

For $S \leq s < \infty$, we have $\Delta V_1(s) \equiv V_1(s) - V_1(s-S)$

$$\begin{aligned} &= \frac{1}{2}\int_{s-S}^{s}[x(\tau) - x_d(\tau)]k_e[x(\tau) - x_d(\tau)]d\tau \\ &\quad -\frac{1}{2}\int_{s-S}^{s}[x(\tau) - x_d(\tau)]k_e[x(\tau - S) - x_d(\tau - S)]d\tau \\ &\quad +\frac{1}{2}\int_{s-S}^{s}[x(\tau) - x_d(\tau)]k_e[x(\tau - S) - x_d(\tau - S)]d\tau \\ &\quad -\frac{1}{2}\int_{s-S}^{s}[x(\tau - S) - x_d(\tau - S)]k_e \times \\ &\quad [x(\tau - S) - x_d(\tau - S)]d\tau \\ &= \frac{1}{2}\int_{s-S}^{s}[x(\tau) - x_d(\tau)]k_e\Delta x(\tau)d\tau \\ &\quad +\frac{1}{2}\int_{s-S}^{s}[x(\tau - S) - x_d(\tau - S)]k_e\Delta x(\tau)d\tau \\ &= \int_{s-S}^{s}[x(\tau) - x_d(\tau) - \frac{1}{2}\Delta x]k_e\Delta x(\tau)d\tau \\ &\leq \int_{s-S}^{s}[k_e(x(\tau) - x_d(\tau))]\Delta x(\tau)d\tau \end{aligned} \tag{27}$$

Note that we have used the property of periodicity $x_d(s+S) = x_d(s)$ and have defined $\Delta x(\tau) = x(\tau) - x(\tau - S)$.

According to Eqs.(16) and (18), we rewrite this inequality as

$$\begin{aligned} \Delta V_1(s) &\leq \int_{s-S}^{s} e_f\Delta x(\tau)d\tau \\ &= \int_{s-S}^{s} e_f(s)\Delta\hat{x}_d(\tau)d\tau \\ &= \int_{s-S}^{s}(-\lambda e_f^2(s) + \frac{1}{v}e_f(s)e_v(s))d\tau \end{aligned} \tag{28}$$

where $\Delta\hat{x}_d(\tau) = \hat{x}_d(\tau) - \hat{x}_d(\tau - S)$.

According to (25), integrating $\bigtriangledown V_2$ from $s - S$ to s and considering Eq.(13), we obtain

$$\Delta V_2 \quad = \int_{s-S}^{s} e_v(s)(-\frac{1}{v}e_f(s) - ce_v(s))d\tau \tag{29}$$

By considering Eqs.(28) and (29), we have

$$\begin{aligned} \Delta V &= \Delta V_1 + \Delta V_2 \\ &\leq \int_{s-S}^{s}(-\lambda e_f^2(s) - ce_v^2(s))d\tau \end{aligned} \tag{30}$$

By looking at Eq.(30), we discuss two cases: $\Delta V < 0$ and $\Delta V = 0$.

Case 1 If $\Delta V < 0$, V is monotonically descending when s increases, which indicates that $\lim_{s\to\infty}V \to 0$ and thus $\lim_{s\to\infty}e \to 0$, $\lim_{s\to\infty}e_v(s) \to 0$ and $\lim_{s\to\infty}x \to x_d$. According to Eq.(2), it yields $\lim_{s\to\infty}e_f(s) \to 0$.

Case 2 If $\Delta V = 0$, since λ and c are positive definite, it leads to $e_f(s) = 0$ and $e_v(s) = 0$ (thus $\bigtriangledown e_v(s) = 0$). Considering the closed-loop dynamics in the space domain (13), we have

$$k_e e(s) = e_f(s) \tag{31}$$

Since $e_f(s) = 0$ and $k_e(s)$ is a positive scalar, we have $e(s) = 0$ and thus $x = x_r$. $e_f(s) = 0$ also indicates that $k_e = 0$ or $x = x_d$ if k_e is nonzero, according to Eq. (16).

By summarizing the results in **Case 1** and **Case 2**, we can conclude that

- $\lim_{s\to\infty} e_v(s) = 0$, $\lim_{s\to\infty} e(s) = 0$ which indicates that the robot's actual trajectory tracks the reference trajectory and its actual velocity tracks the reference velocity.
- $\lim_{s\to\infty} x_r = x_d$ which indicates that the reference trajectory tracks the desired trajectory that generates the desired contact force.
- $\lim_{s\to\infty} e_f(s) = 0$ which indicates that the desired contact force is achieved.

It completes the proof.

IV. Surface exploration

In this section, we verify the validity of the proposed sILC algorithm through simulation studies of a surface exploration task. In particular, the robot gets physical contact with a surface of an unknown shape and it is expected to change its reference trajectory to eventually maintain a desired contact force with the surface. We will first show the features of the proposed algorithm by considering different desired contact forces. Then, we will conduct two comparative simulations to illustrate the advantages of the proposed method by comparing with time-based trajectory learning and impedance control, respectively.

A. sILC Controller

In this simulation of surface exploration, the robot end-effector moves in the $x-y$ plane. Its speed in the x direction is set as $v = 0.1 + \frac{j}{100}$ where the trial number $j = 1, 2, \cdots, 5$. It is worth noting that $\frac{1}{v}$ is included in the learning rate of (17). Therefore, the speed v cannot be 0 and it cannot be too small in practice.For the control parameters in this section, the control effect is better when $v > 0.05m/s$. Therefore,the speed is set to $0.05m/s$ when it is less than $0.05m/s$. The robot end-effector's position in the y direction is subjected to

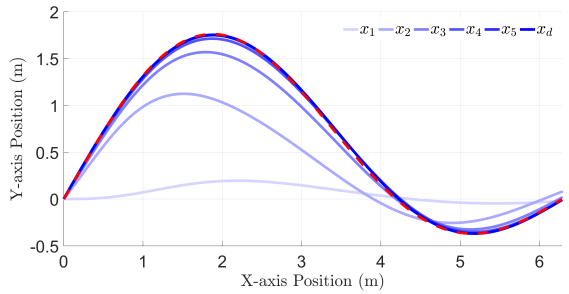

Fig. 2: Surface rest position and robot's actual trajectory in different iterations

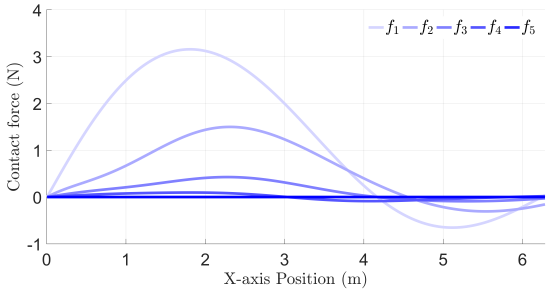

Fig. 3: Contact force in different iterations

a contact force applied by the surface that it interacts with. The desired contact force is specified as $f_d = 0$N, $f_d = 1$N and $f_d = (1 - \frac{x}{2\pi})$N, respectively. As discussed in Section II, the closed-loop dynamics of the robot are described by Eq. (13) where $m = 10, c = 16, k = 19$, respectively. The rest position of the surface is set as $x_d = sinx + sin(\frac{x}{2})$ and its stiffness $k_e = 1$. The learning rate is set as $\lambda = 0.35$.

Simulation results are shown in Figs. 2-3. The trajectory tracking result for the robot with the sILC controller and reference trajectory are shown in Fig. 2. We find that although the rest position of the surface x_d is unknown to the robot, it can be gradually learned by the robot and the tracking error converges to a small neighborhood of zero. The results suggest that the quality of the trajectory improves iteratively with the proposed sILC algorithm. As shown in Fig.3, the contact force between the robot's end-effector and the surface converges to zero after 9 iterations, illustrating that the control objective is achieved. These results also indicate that the robot controller can work effectively and the convergence of the proposed controller is achieved. Importantly, it is noted that the robot's speed in the x direction, i.e., v in each trial is different. As the proposed sILC algorithm is based on the spatial periodicity, it does not require the repeatability of the speed in different trials and thus provides feasibility.

Fig.4 shows the results for the case of $f_d = 1$N. It is found that the robot's actual trajectory deviates from the rest position of the surface in order to produce the desired contact force. As the desired contact force is a constant force, the robot's trajectory is parallel to the rest position of the surface. This is further illustrated by the convergence of the actual contact force to the desired one in the bottom subfigure.

Fig.5 corresponds to the case of $f_d = 1 - \frac{x}{2\times\pi}$ and shows

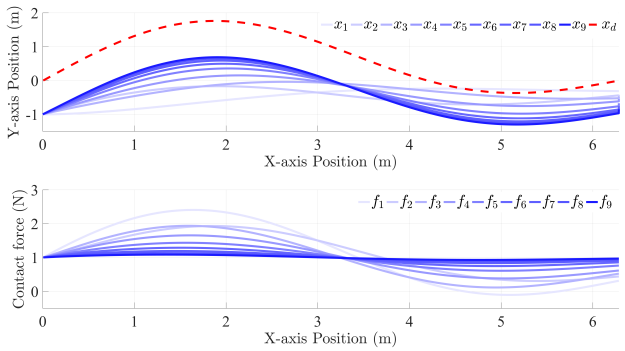

Fig. 4: Path learning results when the desired contact force is constant

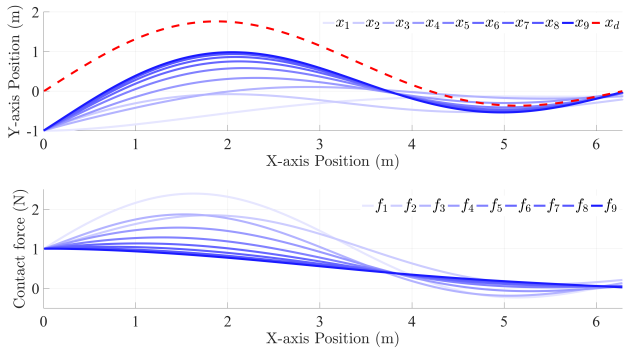

Fig. 5: Path learning results when the desired contact force is time-varying

that the developed algorithm is not only effective for the case where the contact force is constant but also for the case where the contact force is position-varying. In particular, the desired contact force is achieved with a learned path that deviates from the rest position of the surface at the beginning and overlaps at the end. This is useful in a task where a position-varying contact force is needed.

The above results show that after a certain number of iterations, the robot completely learns the desired path to generate a desired contact force. These results are achieved with an initial trajectory of a straight line. Actually, this initial trajectory can be set arbitrarily as long as it does not cause damages to the surface in the learning phase. For example, if partial knowledge of the surface shape is known a priori, the learning can be sped up by setting an initial trajectory close to the unknown desired trajectory.

B. Compare sILC with ILC

In this section, we highlight the superiority of the developed sILC by comparing its performance with the iterative learning control (ILC) method based on time. The ILC method in [3] will be used for this comparison.

The results of trajectory learning using the ILC method based on time are shown in Figs. 6 and 7. For the convenience of observation, these results have been transformed from the time domain to the space domain. Fig. 6 shows a case when

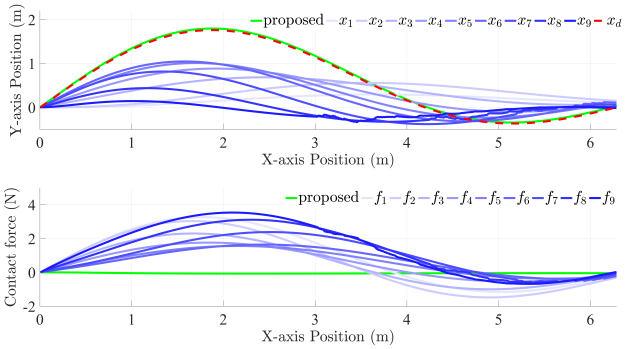

Fig. 6: Time-based trajectory learning performance as the speed increases

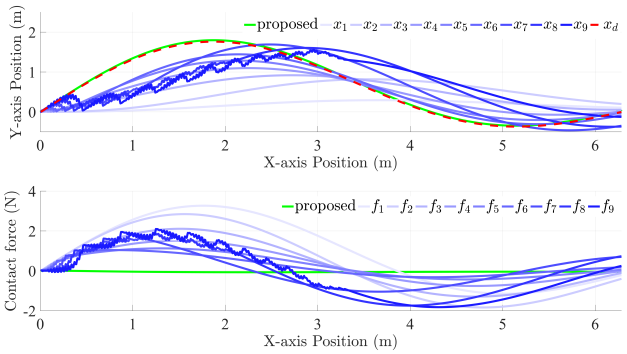

Fig. 7: Time-based trajectory learning performance as the speed decreases

the speed in the x direction increases, i.e., $v(j) = 1 + \frac{j}{50}$ with j the iteration number, while Fig.7 depicts the case when the speed decreases, i.e., $v(j) = 1 - \frac{j}{50}$. It is found that the ILC based on time cannot deal with the case of uncertain speeds in different trials as oscillations and even instability appear. This is because the time period of the learning changes as the speed changes: the positions of the environment where the robot's end-effector gets contact are different at the same time in each iteration. Moreover, the learning result becomes worse when the speed changes more significantly. Therefore, the ILC method based on time can only be used in the case of an iteratively constant speed, while there are many applications where this condition cannot be satisfied due to uncertainties. The proposed sILC method has superior applicability in the sense of addressing this issue. The results of our proposed method after 10 learning iterations are shown as green traces, which show that the actual trajectory (green trace) can approach the desired (dashed red trace) and the contact force (green trace) can approach zero in both cases of speed increase and decrease.

C. Compare sILC with Impedance control

In this section, we conduct a simulation to compare the proposed sILC method with impedance control, which is a widely-used method for robotic interaction tasks.

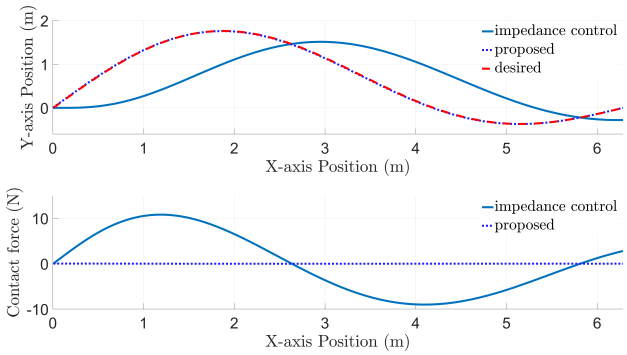

Fig. 8: Comparison between impedance control and the proposed method: robot's trajectory (upper) and contact force (below)

Consider a simple target impedance model with a zero stiffness component:

$$m\ddot{x} + c\dot{x} = k_e(x_0 - x) \tag{32}$$

In order to facilitate comparison of the results, the value settings of inertia and damping matrices are the same as before but the value of k_e is set to $k_e = 5$. By Eq. (32), we can interpret that the robot reacts passively to the contact force exerted by the environment.

The comparison results of impedance control and the proposed sILC method, including the robot's trajectory and contact force, after 10 learning trials are shown in Fig. 8. The upper row of Fig. 8 shows that the desired trajectory can only be achieved by the proposed method. In comparison, the impedance control method leads to a significant tracking error. Correspondingly, the bottom row of Fig. 8 shows that a large contact force is resulted with impedance control, while a zero contact force is achieved with the proposed sILC method.

V. Teaching by Demonstration

In this section, we consider a real-world experimental scenario where a human operator physically contacts with the end effector of a robot and teaches the robot a path to avoid obstacles. In this obstacle avoidance task, we can control the robot in a passive impedance mode and record a trajectory through human teaching. However, this trajectory may not be optimal and needs to be iteratively adjusted by the human until the task requirement is fulfilled. The proposed method allows such adjustments and is tested in the following experiments.

The experimental platform used in this paper is the Sawyer robot shown in Fig. 9, which is comprised of 7-degrees-of-freedom (7-DOF) in a single arm configuration. The Sawyer robot is embedded with several sensors and actuators that are essential for human-robot interaction, including a motor encoder equipped at each joint to measure joint angles and a torque sensor at each joint to measure joint torques [29]. The Sawyer robot runs on the Robot Operating System (ROS) platform. As shown in Fig. 10(a), only the two joints $J1$ and $J3$ of the Sawyer's robot are used. The actual position, velocity and force of the end effector are collected by Sawyer's

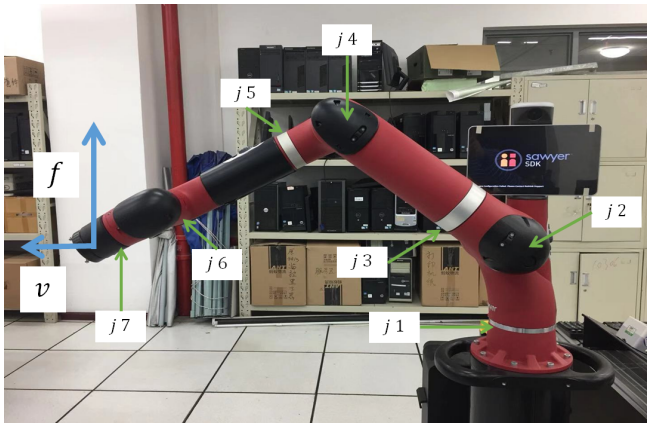

Fig. 9: Experimental platform: Sawyer robot

controller. A laptop with a 2-GHz Intel Core Processor and 4-GB RAM is used to process the collected data and implement the developed method. The laptop communicates with Sawyer (with its own dedicated computer) via TCP. Since the Sawyer robot itself has a position controller, we only need to calculate the target position x_r in the laptop by forward kinematics and transfer it to the robot controller. The axial force is converted by Sawyer's joint torque and can be used after filtering [30].

The parameters in the filter Eq.(15) are set as $m = 1, c = 6, k = 9$. The learning rate is $\lambda = 0.08$. In order to verify that our method works in the case of speed changes, the speed in the horizontal direction is set as $v_j = 0.1 + \frac{j}{100}$ where the trial number $j = 1, 2, \cdots, 7$. The desired contact force $f_d = 0$N.

To illustrate the manipulation skill transfer from the human partner to the robot, the robot is programmed to make a movement in the horizontal direction and the human changes its path by applying forces in the vertical direction. This human-robot interaction process is repeated, so the robot needs to go back to the initial position in each trial and then repeats the movement. The robot stops the learning until the human no longer exerts forces to the robot when they find the robot's trajectory meets their requirement.

Two experiments are carried out to test the learning performance. First, as shown in Fig. 10(b), we assume that the end effector of the robot arm moves from $x = 0.5m$ along a straight line to $x = 0.9m$. When there is an obstacle in the middle, the robot arm learns to avoid the obstacle from the human operator. The white line illustrates the desired trajectory of the human operator, which is determined by the predefined obstacle. The results are shown in Fig. 11, which illustrate that the proposed method is effective in learning human's desired trajectory. In particular, it can be clearly observed that the contact force applied by the human partner is continuously reduced, and the trajectory is gradually adjusted to avoid the obstacle existing on the original trajectory.

In another case, as shown in Fig. 10(c), there are two fixed obstacles between the starting point and the target point, and one of the obstacles can only be avoided from above and the other only from bottom. In this experiment, the trajectory is adjusted from a straight line. Alternatively, the initial trajectory can be obtained by passively following the

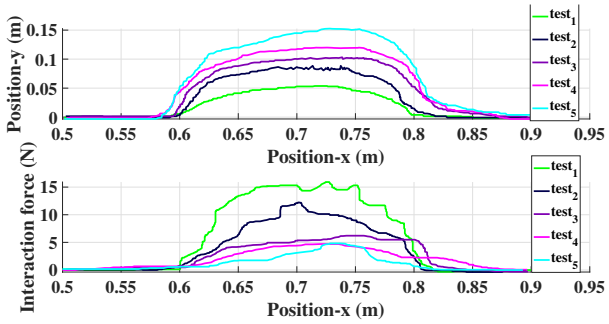

Fig. 11: Robot's trajectories and interaction forces in the scenario of one obstacle

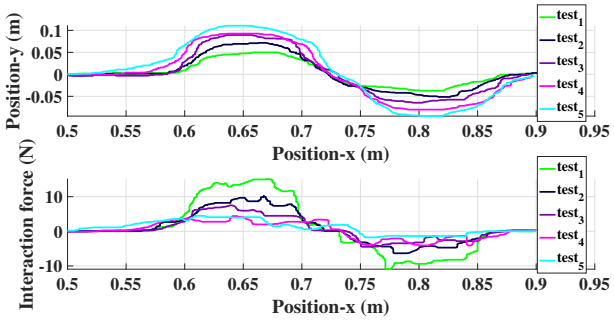

Fig. 12: Robot's trajectories and interaction forces in the scenario of two obstacles

human partner's guidance in zero gravity mode in advance. As can be seen from Fig. 12, after five trials, the robot arm can avoid obstacles by itself without the help of the human operator, which is a result that conventional impedance control cannot achieve. The results also show that with the cooperation of the human operator, the robot can learn arbitrary trajectories by demonstration and there is no limit of a constant speed of the robot in different iterations.

With the change of the number of iterations in the experiment, the average interaction force in each iteration is listed in TABLE I. It can be seen intuitively that the interaction force keeps decreasing, showing the effectiveness of our method.

In [31], a learning method based on the radial basis function neural networks (RBFNN) has been studied to estimate the human partner's intention. For comparison, we use this method to do the same tasks on the Sawyer robot. The number of NN nodes is $p = 8$, and the center of the receptive field $\mu_i = 0$ and the width of the Gaussian function $\eta_i = 1$ for $i = 1, 2, ..., 8$. The adaptation rate is set as $\alpha = 0.005$. The performance of the RBFNN is shown in Figs. 13 and 14, showing oscillations

TABLE I: Average interaction force in each iteration (unit: Newton)

	Experiment 1	Experiment 2
test1	6.47	6.27
test2	4.15	3.76
test3	2.56	2.45
test4	1.89	2.06
test5	1.39	1.75

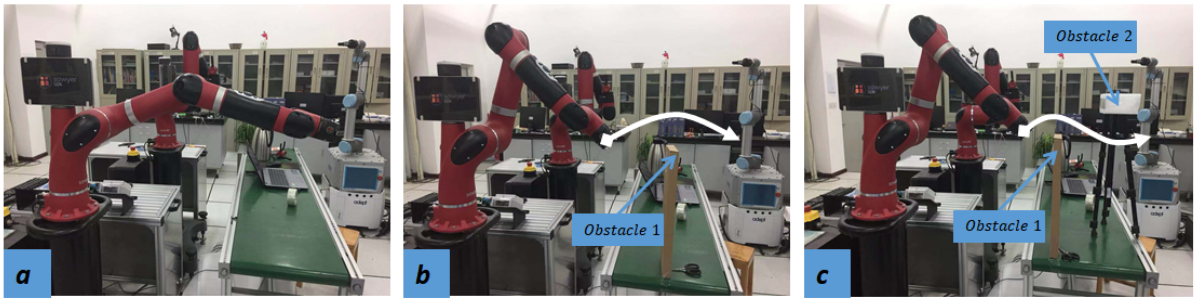

Fig. 10: (a) Experimental setup, (b) Experimental scenario 1 (with one obstacle) and (c) Experimental scenario 2 (with two obstacles)

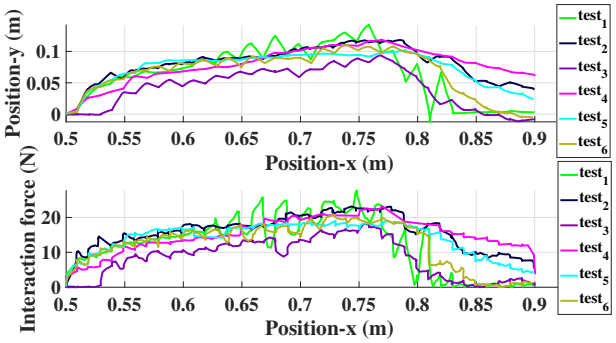

Fig. 13: Robot's trajectories and interaction forces in the scenario of one obstacle under NN based method.

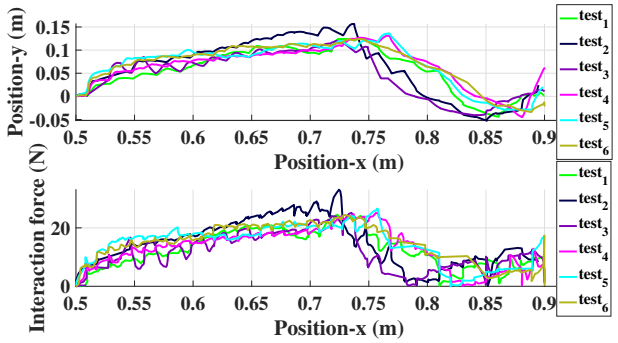

Fig. 14: Robot's trajectories and interaction forces in the scenario of two obstacles under NN based method.

and large contact force. The main reason lies in that the NN based approach updates the path in time, i.e. the interaction forces, position and velocity information is used to update the position at the next time step; if there is a mismatch between the human partner's desired velocity and the robot's reference velocity, either a large interaction force will be generated or the robot cannot move to the human's desired position in time. This phenomenon due to "wrong timing" was also reported in [32]. In comparison, our method updates the path based on the previous iteration, so it is more preferable for repetitive learning tasks. Note that we do not claim our method is more advantageous in all scenarios, as the NN based method is obviously preferred for non-repetitive tasks.

VI. Discussions

This paper addresses the problem of path learning based on spatial iterative learning control. We introduced a novel sILC method into path learning, making it possible to work effectively with uncertain speeds in each trial. Also, we have shown that this method works well even if the desired contact force is time-varying. The stability and convergence of this novel sILC method have been rigorously analyzed based on the Lyapunov theory. Using simulations, we have shown that path learning enables maintenance of contact force during tasks like surface exploration and prevents the robot from applying high forces.

A human-robot interaction example has been considered in the experiments, which have shown that the proposed method allows the robot to learn a desired path from the human operator. The robot initially followed the human in a passive manner and then iteratively adjusted its planned path according to the proposed learning method. It led to the robot's increasing contribution to the task and correspondingly decreasing human effort, so the desired manipulation path was eventually transferred from the human to the robot. The proposed method addresses an important problem of varying and uncertain human movement speeds in different trials, thanks to its property of spatial iterative learning, so it provides flexibility to physical human-robot interaction.

The method proposed in this article only shows the results in the $X-Y$ plane. In fact, the method proposed in this paper can be used in more dimensional spaces (such as $X-Y-Z$). In this case, iterative algorithm in the Y direction can be added to apply a force on the Z axis to change the position on the Z axis.

Future works will focus on testing the proposed method in a more complicated scenario, e.g. with a movement in multiple directions. Surface exploration on a workpiece based on the sILC will be also investigated in a real-world setup.

References

[1] Y. Maeda, T. Hara, and T. Arai, "Human-robot cooperative manipulation with motion estimation," in *IEEE Conference on Intelligent Robots and Systems (IROS 2001)*, vol. 1, (Maui), pp. 2240–2245, October 29-November 29 2001.

[2] Y. Li and S. S. Ge, "Impedance learning for robots interacting with unknown environments," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1422–1432, 2014.

[3] Y. Li and S. S. Ge, "Force tracking control for motion synchronization in humancrobot collaboration," *Robotica*, vol. 34, no. 6, pp. 1260–1281, 2016.

[4] Z. Liu, C. Chen, and Y. Zhang, "Adaptive neural control for dual-arm coordination of humanoid robot with unknown nonlinearities in output mechanism," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 521–532, 2015.

[5] S. Jung, T. C. Hsia, and R. G. Bonitz, "Force tracking impedance control of robot manipulators under unknown environment," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 474–483, 2004.

[6] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 918–930, 2011.

[7] Y. Li, G. Ganesh, N. Jarrassé, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Force, impedance, and trajectory learning for contact tooling and haptic identification," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1170–1182, 2018.

[8] H. I. H. Asada, "Automatic program generation from teaching data for the hybrid control of robots," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 166–173, 1989.

[9] F. Matthew, S. David, and P. Zengxi, "Learning trajectories for robot programing by demonstration using a coordinated mixture of factor analyzers," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 706–717, 2016.

[10] R. Zollner, O. Rogalla, R. Dillmann, and M. Zollner, "Understanding users intention: Programming fine manipulation tasks by demonstration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, vol. 1, (Lausanne), pp. 1114–1119, September 30-October 4 2002.

[11] L. Lukic, A. Billard, and J. Santos-Victor, "Motor-primed visual attention for humanoid robots," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 2, pp. 76–91, 2015.

[12] J. Aleotti and S. Caselli, "Robust trajectory learning and approximation for robot programming by demonstration," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 54, (Barcelona), pp. 409–413, April 18-22 2006.

[13] J. Aleotti, G. Micconi, and S. Caselli, "Programming manipulation tasks by demonstration in visuo-haptic augmented reality," in *IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, (Richardson), pp. 13–18, October 10-11 2014.

[14] A. Duschau-Wicke, J. von Zitzewitz, A. Caprez, L. Lunenburger, and R. Riener, "Path control: A method for patient-cooperative robot-aided gait rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 1, pp. 38–48, 2010.

[15] B. Hwang, B.-M. Oh, and D. Jeon, "An optimal method of training the specific lower limb muscle group using an exoskeletal robot," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 4, pp. 830–838, 2018.

[16] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Anchorage), pp. 2074–2081, May 3-8 2010.

[17] C. G. T. Panagiota, M. Sotiris, "Human-robot interaction review and challenges on task planning and programming," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 8, pp. 916–931, 2016.

[18] A. A. E. Osman, R. A. El-Khoribi, and M. E. Shoman, "Trajectory learning using principal component analysis," in *World Conference on Information Systems and Technologies (WorldCIST)*, vol. 1, (Madeira), pp. 174–183, April 11-13 2017.

[19] M. I. V. Aleksandar and I. Andrew, "Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 42, no. 4, pp. 1039–1052, 2012.

[20] M. Ogino, H. Toichi, Y. Yoshikawa, and M. Asada, "Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping," in *World Conference on Information Systems and Technologies (WorldCIST)*, vol. 54, (Barcelona), pp. 414–418, April 18-22 2006.

[21] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.

[22] S. Marjorie and A. V. Richard, "Acquiring robust, force-based assembly skills from human demonstration," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 772–782, 2000.

[23] Y. Jian, D. Huang, J. Liu, and D. Min, "High-precision tracking of piezoelectric actuator using iterative learning control and direct inverse compensation of hysteresis," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 368–377, 2019.

[24] J.-X. Xu and D. Huang, "Spatial periodic adaptive control for rotary machine systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2402–2408, 2008.

[25] C. Yang, K. Huang, H. Cheng, Y. Li, and C. Su, "Haptic identification by elm-controlled uncertain manipulator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp. 2398–2409, 2017.

[26] D. Lee and C. Ott, "Incremental kinesthetic teaching of motion primitives using the motion refinement tube," *Autonomous Robots*, vol. 31, no. 2, pp. 115–131, 2011.

[27] J. J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.

[28] N. Hogan, "Impedance control: an approach to manipulation-Part I: Theory; Part II: Implementation; Part III: Applications," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–24, 1985.

[29] U. E. Ogenyi, G. Zhang, and C. Yang, "An intuitive robot learning from human demonstration," in *Intelligent Robotics and Applications. 11th International Conference (ICIRA)*, (Newcastle), pp. 176–185, August 9-11 2018.

[30] Y. Park, S. C. Ryu, R. J. Black, K. K. Chau, B. Moslehi, and M. R. Cutkosky, "Exoskeletal force-sensing end-effectors with embedded optical fiber-bragg-grating sensors," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1319–1331, 2009.

[31] Y. Li and S. S. Ge, "Humancrobot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2014.

[32] D. P. Losey and M. K. O'Malley, "Learning the correct robot trajectory in real-time from physical human interactions," *ACM Transactions on Human-Robot Interaction*, vol. 9, no. 1, pp. 1–19, 2019.